

IMAGE CODER

Publication number: JP8102938

Publication date: 1996-04-16

Inventor: MIHASHI SATOSHI; HIRONO CHIHARU

Applicant: SONY CORP

Classification:

- International: **H04N7/26; G06T9/00; H03M7/36; H04N7/24; H04N7/26; G06T9/00; H03M7/36; H04N7/24; (IPC1-7): H04N7/24; G06T9/00; H03M7/36**

- European:

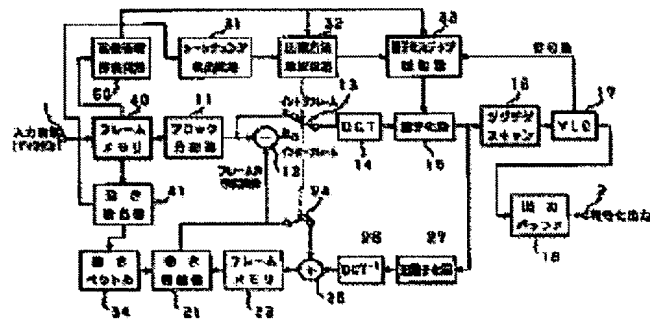
Application number: JP19940238110 19940930

Priority number(s): JP19940238110 19940930

Report a data error here

Abstract of JP8102938

PURPOSE: To improve image quality by providing an image data storage means, a scene change detection means, a compression method selection means, a compression coding means and an output buffer means to the coder to implement efficient image compression. **CONSTITUTION:** Image data stored in a frame memory 40 are sent properly to an image information evaluation circuit 50. The circuit 50 calculates a parameter representing an image information quantity and enough to estimate a data quantity after compression. A compression method selection circuit 32 selects any of the compression systems of in-frame coding/inter-frame prediction coding based on the information quantity parameter from the circuit 50 and an output from a scene change detection circuit 31. A motion detector 41 and a motion vector generating circuit 34 detect a motion vector required for compression by the in-frame coding. A quantization step controller 33 controls sufficiently preceding frames by a quantization step of a quantizer 15 so as to compress the frames by the in-frame coding thereby reducing a residual data amount in an output buffer 18.



Data supplied from the esp@cenet database - Worldwide

(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平8-102938

(43)公開日 平成8年(1996)4月16日

(51)Int.Cl.⁶

識別記号

庁内整理番号

F I

技術表示箇所

H 0 4 N 7/24

G 0 6 T 9/00

H 0 3 M 7/36

9382-5K

H 0 4 N 7/ 13

Z

G 0 6 F 15/ 66

3 3 0 J

審査請求 未請求 請求項の数8 O L (全 18 頁)

(21)出願番号

特願平6-238110

(22)出願日

平成6年(1994)9月30日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 三橋 聡

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(72)発明者 広野 千春

東京都品川区北品川6丁目7番35号 ソニー株式会社内

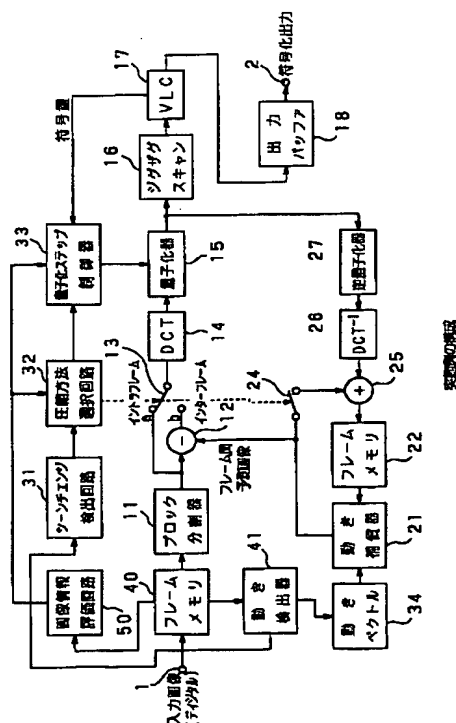
(74)代理人 弁理士 小池 晃 (外2名)

(54)【発明の名称】 画像符号化装置

(57)【要約】

【構成】 複数枚フレーム分のフレームメモリ40と、メモリ40に蓄積された画像データをブロック分割し各ブロック毎にサーチ側とリファレンス側のピクセル毎に求めた誤差値を1画面分合計した合計値と、時間的に過去の1画面分の誤差値を合計した合計値との比に基づいてシーンチェンジ検出するシーンチェンジ検出回路31と、その検出出力に応じて画像データの圧縮方法を選択する圧縮方法選択回路32と、選択した圧縮方法に従って画像データを圧縮符号化する差分器12から可変長符号化回路17まで及びローカルデコーダと、圧縮符号化された画像データを蓄積する出力バッファ18と、シーンチェンジ検出出力に応じて出力バッファ18の蓄積量を制御する量子化ステップ制御器33とを有する。

【効果】 シーンチェンジが検出でき、効率の良い画像圧縮が可能で、全体的に画質を向上させることができると共に、コストの低減も可能である。



(2)

1

【特許請求の範囲】

【請求項1】 入力画像データを複数枚蓄える画像データ蓄積手段と、

上記画像データ蓄積手段に蓄積された入力画像データを複数のブロックに分割し、それぞれのブロック毎にサーチ側とリファレンス側のピクセル毎に求めた誤差値を1画面分合計した第1の合計値と、時間的に過去の1画面分の誤差値を合計した第2の合計値との比に基づいてシーンチェンジ検出を行うシーンチェンジ検出手段と、

上記シーンチェンジ検出手段によるシーンチェンジ検出出力に応じて画像データの圧縮方法を選択する圧縮方法選択手段と、

上記圧縮方法選択手段により選択された圧縮方法に従って上記入力画像データに所定の圧縮符号化処理を施す圧縮符号化処理手段と、

圧縮符号化された画像データを蓄積する出力バッファ手段と、

上記シーンチェンジ検出手段によるシーンチェンジ検出出力に応じて上記出力バッファ手段の蓄積量を制御するバッファ制御手段とを有することを特徴とする画像符号化装置。

【請求項2】 上記シーンチェンジ検出手段は、上記第1の合計値が所定の閾値以下であるときには上記第2の合計値を保存しておくことを特徴とする請求項1記載の画像符号化装置。

【請求項3】 上記シーンチェンジ検出手段は、シーンチェンジを検出したときに上記第2の合計値を保存しておくことを特徴とする請求項1記載の画像符号化装置。

【請求項4】 上記シーンチェンジ検出手段は、上記第1の合計値が所定の閾値以下であるとき、及び、シーンチェンジを検出したときに上記第2の合計値を保存しておくことを特徴とする請求項1記載の画像符号化装置。

【請求項5】 入力画像データを複数枚蓄える画像データ蓄積手段と、

上記画像データ蓄積手段に蓄積された入力画像データを複数のブロックに分割し、それぞれのブロック毎にサーチ側とリファレンス側のピクセル毎に求めた差分値の第1の絶対値和と、時間的に過去の差分値の第2の絶対値和との比に基づいてシーンチェンジ検出を行う第1のシーンチェンジ検出手段と、

上記第1のシーンチェンジ検出手段でシーンチェンジと判定したブロックの1画面に対する割合に基づいて、当該1画面のシーンチェンジを検出する第2のシーンチェンジ検出手段と、

上記第2のシーンチェンジ検出手段によるシーンチェンジ検出出力に応じて1画面の画像データの圧縮方法を選択する圧縮方法選択手段と、

上記圧縮方法選択手段により選択された圧縮方法に従って上記入力画像データに所定の圧縮符号化処理を施す圧縮符号化処理手段と、

2

圧縮符号化された画像データを蓄積する出力バッファ手段と、

上記シーンチェンジ検出手段によるシーンチェンジ検出出力に応じて上記出力バッファ手段の蓄積量を制御するバッファ制御手段とを有することを特徴とする画像符号化装置。

【請求項6】 上記第1のシーンチェンジ検出手段は、上記第1の絶対値和が所定の閾値以下であるときには上記第2の絶対値和を保存しておくことを特徴とする請求項5記載の画像符号化装置。

【請求項7】 上記第1のシーンチェンジ検出手段は、シーンチェンジを検出したときに上記第2の絶対値和を保存しておくことを特徴とする請求項5記載の画像符号化装置。

【請求項8】 上記第1のシーンチェンジ検出手段は、上記第1の絶対値和が所定の閾値以下であるとき、及び、シーンチェンジを検出したときに上記第2の絶対値和を保存しておくことを特徴とする請求項5記載の画像符号化装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、例えば画像を圧縮符号化する場合に用いて好適な画像符号化装置に関する。

【0002】

【従来の技術】 従来の例えば画像を圧縮符号化する場合に用いて好適な画像符号化装置の構成例を図9に示す。この図9の画像符号化装置において、入力端子1には、図10に示すように、

輝度成分(Y) 352(H)×240(V)×30フレーム

クロマ成分(Cb) 174(H)×120(V)×30フレーム

クロマ成分(Cr) 174(H)×120(V)×30フレーム

のピクセル数にデジタル化された画像データが供給される。

【0003】 上記入力端子1に供給された入力画像データは、当該入力画像データを一時的に蓄えて然るべき順番に入れ替えるためのフレームメモリ10を介して、動き検出器20とブロック分割器11に送られる。当該ブロック分割器11は、フレームメモリ10から供給されたそれぞれのフレームを、図11に示すように、輝度成分(Y)、クロマ成分(Cr)、(Cb) それぞれを8×8ピクセルのブロックに分割する。なお、輝度成分(Y)の4つのブロック(Y0, Y1, Y2, Y3)と1つのクロマ成分(Cb)のブロックと、1つのクロマ成分(Cr)のブロックからなる合計6つのブロック(Y0, Y1, Y2, Y3, Cb, Cr)は、マクロブロック(MB)と呼ばれている。なお、マクロブロックを構成する輝度画素(ルミナンスピクセル)とクロマ画素(クロミナンスピクセル)との関係は、図12に示すようになっている。

【0004】 このブロック分割器11からのマクロブロック単位のデータは差分器12に送られる。この差分器

(3)

3

12では、ブロック分割器11からのデータと後述するフレーム間予測画像データとの差分をとり、その出力を後述するフレーム間予測符号化がなされるフレームのデータとして切換スイッチ13の被切換端子bに送る。また、当該切換スイッチ13の被切換端子aには、上記ブロック分割器11からのデータが後述するフレーム内符号化がなされるフレームのデータとして供給される。

【0005】上記切換スイッチ13を介したブロック単位のデータはDCT回路14によって離散コサイン変換(DCT)処理され、そのDCT係数が量子化器15に送られる。当該量子化器15では、所定の量子化ステップ幅で上記DCT出力を量子化し、この量子化した係数がジグザグスキャン回路16に送られる。当該ジグザグスキャン回路16では、上記量子化係数を図13に示すようにいわゆるジグザグスキャンによって並べ換え、その出力を可変長符号化回路17に送る。この可変長符号化回路17では、上記ジグザグスキャン回路16の出力データを可変長符号化(VLC)し、その出力を出力バッファ18に送ると共に、当該可変長符号化処理により発生した符号量を示す情報を、量子化ステップ制御器19に送る。量子化ステップ制御器19は、可変長符号化回路17からの符号量を示す情報に基づいて量子化器15の量子化ステップ幅を制御する。また、上記出力バッファ18から出力されたデータは圧縮符号化がなされた符号化出力として出力端子2から出力される。

【0006】また、上記量子化器15からの出力は、逆量子化器27によって逆量子化され、さらに逆DCT回路26によって逆DCT処理される。当該逆DCT回路26の出力は、加算器25に送られる。この加算器25には、フレーム間予測符号化のフレームのときにオンとなる切換スイッチ24を介した動き補償器21からのフレーム間予測画像データも供給され、当該データと上記逆DCT回路26の出力データとの加算が行われる。この加算器25の出力データは、フレームメモリ22に一時的に蓄えられた後、動き補償器21に送られる。

【0007】当該動き補償器21は、上記動き検出器20によって検出された動きベクトルに基づいて動き補償を行い、これによって得たフレーム間予測画像データを出力する。以下、上記図9の従来の画像符号化装置の具体的な動作について詳細に説明する。ここで、説明のために以下のように各フレームの呼び名を定義する。

【0008】まず、表示順にフレームを並べたとき、それぞれをI0、B1、B2、P3、B4、B5、P6、B7、B8、I9、B10、B11、B12、・・・と呼ぶこととする。これらのフレームのうち、I、P、Bは、後に説明するが、圧縮方法の種類を示し、これらI、P、Bの次の数字は、単純に表示順を示している。また、フレーム間予測符号化におけるこれらI、P、Bの関係は、図14に示すようになる。

【0009】カラー動画像符号化方式の国際標準化作業

4

グループであるいわゆるMPEG (Moving Picture Expert Group)のうちMPEG1では、この様な画像を圧縮するために、以下のようにすることが規定されている。まず、I0の画像を圧縮する。次に、P3の画像を圧縮するのだが、P3そのものを圧縮するのではなく、P3とI0の画像との差分データを圧縮する。

【0010】その次に、B1の画像を圧縮するのだが、B1そのものを圧縮するのではなく、B1とI0或いは、B1とP3との差分データ或いはI0とP3の平均値との差分(いずれか情報の少ない方)を圧縮する。その次に、B2の画像を圧縮するのだが、B2そのものを圧縮するのではなく、B2とI0或いは、B2とP3との差分データ或いはI0とP3の平均値との差分(どちらか情報の少ない方を選んで)を圧縮する。

【0011】次に、P6の画像を圧縮するのだが、P6そのものを圧縮するのではなく、P6とP3の画像との差分データを圧縮する。上述したような処理を順番に並べて表すと、

	処理対象の画像	差分をとる相手
(1)	I0	—
(2)	P3	I0
(3)	B1	I0又はP3
(4)	B2	I0又はP3
(5)	P6	P3
(6)	B4	P3又はP6
(7)	B5	P3又はP6
(8)	P9	P6
(9)	B7	P6又はP9
(10)	B8	P6又はP9
(11)	I9	—
(12)	P12	I0
(13)	B10	I9又はP12
(14)	B11	I9又はP12

となる。このようにエンコード順は、I0、P3、B1、B2、P6、B4、B5、P9、B7、B8、I9、P12、B10、B11、・・・のように、表示順とは順番が入れ替わる。圧縮後のデータ(符号化データ)はこの順番に並ぶことになる。

【0012】以下、上述したことを図9の構成の動作と共にさらに詳しく述べる。1枚目の画像(すなわちI0)のエンコードでは、まず、上記フレームメモリ10から1枚目に圧縮すべき画像のデータが、ブロック分割器11によってブロック化される。このブロック分割器11からは、前記Y0、Y1、Y2、Y3、Cb、Crの順にブロック毎のデータが出力され、被切換端子a側に切り換えられた切換スイッチ13を介してDCT回路14に送られる。このDCT回路14では、それぞれの

(4)

5

ブロックについて縦横2次元の離散コサイン変換処理を行う。これにより、時間軸であったデータが周波数軸に変換される。

【0013】このDCT回路14からのDCT係数は、量子化器15に送られ、当該量子化器15で所定の量子化ステップ幅で量子化される。その後、ジグザグスキャン回路16によって図13のようにジグザグ順に並べ変えられる。このようにジグザグ順に並べると、後ろへ行くほど、その係数は周波数成分の高い係数となるから、一般的に係数の値は後ろの方が小さくなる傾向にある。したがって、ある値Sで量子化すると、後ろへ行くほど、その結果は0になる頻度が増し、結果的に高域の成分が切り落とされることになる。

【0014】その後、この量子化後の係数は、可変長符号化(VLC)回路17へ送られ、ここでいわゆるハフマンコーディングが施される。この結果得られる圧縮されたビットストリームは、出力バッファ18に一旦蓄えられた後、一定のビットレートで送出される。当該出力バッファ18は、不規則に発生するビットストリームを一定のビットレートで送出できるようにするための緩衝のためのメモリである。

【0015】以上の様に1枚の画像だけ単独で圧縮することをフレーム内(イントラ: Intra)符号化と言い、この画像をIピクチャと呼ぶ。したがって、デコーダが上記のIピクチャのビットストリームを受信した場合は、以上に述べたことを逆にたどり、1枚目の画像を完成させる。次に、2枚目の画像(すなわちP3)のエンコードでは、以下のようになされる。

【0016】すなわち、この2枚目以降もIピクチャとして圧縮してビットストリームを作っても良いが圧縮率を上げるには、連続する画像の内容には相関があることを利用して、以下の様な方法で圧縮する。まず、動き検出器20では、2枚目の画像を構成するマクロブロック毎に、1枚目の画像(I0)の中からそれに良く似たパターンを捜し出し、それを動きベクトルという(x, y)の相対位置の座標として表現する。

【0017】また、2枚目の画像ではそれぞれのブロックを、上記Iピクチャの場合のようにそのままDCT回路14に送るのではなく、そのブロック毎の動きベクトルに従って一枚目の画像から引っ張ってきたブロックとの差分のデータ(差分器12による差分データ)を、DCT回路14へ送るようにする。なお、動きベクトルの検出方法としては、ISO/IEC 11172-2 annex D. 6.2 に詳細に述べられているためここでは省略する。

【0018】ここで、例えば上記動きベクトルによって示された一枚目の画像のパターンと、これから圧縮しようとするブロックのパターンとの間で、相関が非常に強くなっていれば、その差分データは非常に小さくなり、したがって、上記フレーム内(イントラ)符号化で圧縮するよりも、上記動きベクトルと上記差分データとを符

6

号化した方が、圧縮後のデータ量は小さくなる。

【0019】このような圧縮方法を、フレーム間(インター: Inter)予測符号化と呼んでいる。ただし、常に差分データが少なくなるわけではなく、絵柄(画像内容)によっては、差分を取るよりも、上記フレーム内符号化で圧縮した方が、圧縮率が上がる場合がある。このような場合は、上記フレーム内符号化で圧縮する。フレーム間予測符号化にするか、フレーム内符号化にするかは、マクロブロック毎に異なる。

10 【0020】以上のことを図9の画像符号化装置(エンコーダ)に即して説明すると、まず、フレーム間予測符号化を行うためには、エンコーダ側でたえずデコーダ側で作られる画像と同じ画像を作っておく必要がある。そのためにエンコーダ内には、デコーダと同じ回路が存在する。その回路をローカルデコーダ(局部復号器)と呼ぶ。図9の逆量子化器27と逆DCT回路26と加算器25とフレームメモリ22と動き補償器21が当該ローカルデコーダに対応し、フレームメモリ22内に記憶される画像のことをローカルデコードドピクチャ(Local decoded picture)又はローカルデコードドデータ(Local decoded data)と呼ぶ。これに対して、圧縮前の画像のデータは、オリジナルピクチャ(Original picture)又はオリジナルデータ(Original data)と呼ぶ。

【0021】なお、前述した1枚目のIピクチャの圧縮時にも、上記ローカルデコーダを通して復号化された1枚目の画像が、上記フレームメモリ22内に格納される。ここで、注意すべきことは、このローカルデコーダによって得られる画像は、圧縮前の画像ではなく、圧縮後復元した画像であり、圧縮による画質劣化のある、デコーダが復号化する画像とまったく同じ画像であるということである。

【0022】このような状態のエンコーダに2枚目の画像(P3)のデータ(Original data)が入ってくるわけだが(この段階ですでに、動きベクトルは検出済でなければならない)、データはブロック毎に動きベクトルを持ち、このベクトルが動き補償器(MC: Motion Compensation)21に与えられる。当該動き補償回路21は、その動きベクトルの示すローカルデコードドピクチャ上のデータ(動き補償データ: MC data: 1マクロブロック)を上記フレーム間予測画像データとして出力する。

40 【0023】上記2枚目のオリジナルデータとこの動き補償データ(フレーム間予測画像データ)のピクセル毎の、差分器12による差分データが、上記DCT回路14に入力される。それからの後の圧縮方法は、基本的にIピクチャと同じである。上述のような圧縮方法によって圧縮する画像をPピクチャ(Predicted picture)と呼ぶ。

50 【0024】さらに詳しく説明すると、Pピクチャにおいてすべてのマクロブロックがフレーム間予測符号化で圧縮するとは限らず、フレーム内符号化で圧縮する方が

(5)

7

効率が良いと判断されるときは、そのマクロブロックは当該フレーム内符号化で符号化を行う。すなわち、Pピクチャにおいても、マクロブロック毎に、フレーム内符号化によるか（このマクロブロックをイントラマクロブロックと呼ぶ）、又はフレーム間予測符号化によるか（このマクロブロックをインターマクロブロックと呼ぶ）のどちらかを選択して圧縮を行う。

【0025】上述のように、上記ローカルデコーダでは、量子化器15の出力が、逆量子化器27で逆量子化され、さらに逆DCT回路26で逆DCT処理された後、エンコード時に動き補償データ（MC data）と足され最終的なローカルデコードピクチャとなる。次に、3枚目の画像（すなわちB1）のエンコードでは、以下のようになされる。

【0026】上記3枚目の画像（B1）のエンコードでは、I0、P3の2枚の画像それぞれに対する動きベクトルを探索する。ここで、I0に対する動きベクトルをフォワードベクトル（forward Vector）MVf(x,y)と呼び、P3に対する動きベクトルをバックワードベクトル（Backward Vector）MVb(x,y)と呼ぶ。この3枚目の画像についても差分データを圧縮するわけであるが、どのデータを圧縮するのかが、問題である。この場合も一番情報量が少なくなるものとの差分を取れば良い。このときの圧縮方法の選択肢としては、

（1）フォワードベクトルMVf(x,y)の示すI0上のデータとの差分

（2）バックワードベクトルMVb(x,y)の示すP3上のデータとの差分

（3）フォワードベクトルMVf(x,y)の示すI0上のデータとバックワードベクトルMVb(x,y)の示すP3上のデータの平均値との差分

（4）差分データは使わない（フレーム内符号化）の4つである。マクロブロック毎にこの4種類の圧縮方法から一つが選択される。上記圧縮方法の選択肢のうちの（1）、（2）、（3）の場合は、それぞれの動きベクトルも動き補償器21に送られ、差分器21ではその動き補償データとの差分をとり、これがDCT回路14に送られる。上記圧縮方法の選択肢のうちの（4）の場合は、そのままのデータがDCT回路14へ送られる。

【0027】上述した1枚目、2枚目のエンコードの処理の結果、ローカルデコードピクチャを格納するフレームメモリ22には、I0、P3の2枚のピクチャが、復元されているのでこのようなことが可能である。次に、4枚目の画像（すなわちB2）のエンコードでは、以下のようになされる。

【0028】上記4枚目の画像（B2）のエンコードでは、上述した3枚目（B1）のエンコード方法のところで、B1をB2に置き換えたこと以外は、上記3枚目のエンコードと同じ方法で圧縮する。次に、5枚目の画像（すなわちP6）のエンコードでは、以下のよ

8

うになされる。

【0029】上記5枚目の画像（P6）のエンコードでは、上述した2枚目（P3）のエンコード方法のところで、P3をP6に、I0をP3に置き換えただけで、他は同じ説明となる。6枚目以降は、上述の繰り返しとなるので説明は省略する。また、MPEGにおいては、GOP（Group Of Picture）と呼ばれるものが規定されている。

【0030】すなわち、何枚かのピクチャの集まりがグループオブピクチャ（GOP）と呼ばれており、当該GOPは符号化データ（圧縮後のデータ）上で見て連続した画像の集まりでなくてはならないものである。また、GOPはランダムアクセスを考慮したもので、そのためには符号化データ上で見てGOPの最初に来るピクチャは上記Iピクチャである必要がある。さらに、表示順（ディスプレイ順）でGOPの最後は、I又はPピクチャでなくてはならない。

【0031】図15には、最初が4枚のピクチャからなるGOPで、それ以降が6枚のピクチャからなるGOPとなる例を挙げる。なお、図15のAはディスプレイ順を示し、図15のBは符号化データ順を示している。この図15において、GOP2に注目すると、B4、B5はP3、I6から形成されるため、例えばランダムアクセスでI6にアクセスされると、P3が無い場合、B4、B5は正しく復号化できない。このようにGOP内だけで正しく復号化できないGOPは、クローズドGOP（Closed GOP）でないという。

【0032】これに対し、もしB4、B5がI6だけしか参照していないとしならば、例えばランダムアクセスでI6にアクセスしても、P3は必要ないため、これらB4、B5は正しく復号化できることになる。このようにGOP内だけの情報で、完全に復号化できるGOPをクローズドGOP（Closed GOP）と呼ぶ。以上のような圧縮方法の選択の中から一番効率の良い方法で圧縮するわけであるが、その結果発生する符号化データ（Coded data）の量は、入力画像にも依存し、実際に圧縮してみないと判らない。

【0033】しかし、圧縮後のデータのビットレートを一定にするためにコントロールすることも必要である。当該コントロールを行うためのパラメータは、量子化器15に与える前記符号量を表す情報としての量子化ステップ（又は量子化スケール：Q-scale）である。同じ圧縮方法でも、上記量子化ステップを大きくすれば発生ビット量は減り、小さくすれば増える。

【0034】この量子化ステップの値は、次のようにして制御する。エンコーダには、圧縮後のデータを一定のビットレートにするために、出力に緩衝バッファ（出力バッファ18）が設けられており、これによって画像毎のある程度のデータ発生量の差は吸収できるようになされている。しかし、定められたビットレートを超えるよ

9

うなデータの発生が続けば、出力バッファ18の残量が増加し、ついにはオーバーフローを起こすことになる。逆にビットレートを下回るデータの発生が続けば出力バッファ18の残量は減少し、最後にはアンダーフローを引き起こすことになる。

【0035】したがって、エンコーダは、出力バッファ18の残量をフィードバックすることにより、前記量子化ステップ制御器19が量子化器15の量子化ステップをコントロールし、ここで出力バッファ18の残量が少なくなればあまり圧縮しないように量子化ステップを小さくするよう制御し、出力バッファ18の残量が多くなれば圧縮率を高くするように量子化ステップを大きくするようにコントロールを行うようにしている。

【0036】また、前述した圧縮方法（前記フレーム内符号化やフレーム間予測符号化）によって発生する符号化データ量の範囲には、大きな差がある。特にフレーム内符号化方式で圧縮をすると大量のデータが発生するため、出力バッファ18の空き容量が小さい場合には量子化ステップサイズを大きくしなければならず、場合によっては量子化ステップサイズを最大にしてもバッファ18のオーバーフローを招くかもしれない。よしんばバッファ18に収まったとしても量子化ステップが大きければフレーム内符号化の画像は後のフレーム間予測符号化の画質に影響するので、フレーム内符号化での圧縮を行う前には出力バッファ18に十分な空き容量が必要である。

【0037】したがって、予め定められた順序の圧縮方法を決めておき、フレーム内符号化の前には十分な出力バッファ18の空き容量を確保するように、量子化ステップ制御器19は量子化ステップサイズのフィードバックコントロールを行うようにしている。以上のようにして一定レートの符号化データに抑えることが可能となる。

【0038】

【発明が解決しようとする課題】ところで、上述した従来の方法では、以下の理由により高画質を得られないことが欠点となっている。すなわち、出力バッファの残量をフィードバックすることによって量子化ステップをコントロールする従来の方法では、フィードバックに基本的に遅延があるため入力画像の急激な情報量変化に追従できない。そのため、入力画像の情報量が急激に増加した場合、大量の圧縮データが発生し、バッファの残量を大幅に減らし、時にはオーバーフローを引き起こす。また、オーバーフローしなかったとしても、出力バッファの残量は非常に少なくなっているため、次に圧縮される画像は量子化ステップが大きくなり、その結果画質の低下を招くことになる。逆に、入力画像の情報量が急激に減少した場合、発生する圧縮データは小さなものになるにもかかわらず、上記遅延のために量子化ステップはすぐには小さくならないので、当該遅延の間の画質を上げる

(6)

10

ことができない。

【0039】また、一般に入力画像は様々で、同じ圧縮率であっても同じ画質を得られるとは限らない。それは画像の情報量そのものの違いや、圧縮方法に依存した特徴（前後で相関が高い等）に依存するからである。したがって、画質を均質に圧縮するには画像に適応的に圧縮方法、圧縮率などを変えなければならない。さらに、時々刻々情報量の変化する入力画像を一定のビットレートで平均的に高画質に圧縮するためには、出力バッファによって低ビットレートを維持できる範囲でかつ画質が均質になるように、情報量の多い画像（絵）には多めの圧縮データを許し、情報量の少ない画像には少なめの圧縮データにすることが必要だが、次のような場合に従来の方法ではそれができない。

【0040】例えば、情報量の少ない画像が連続し、そのあとで急に情報量の多い画像が入ってくる場合を考えると、先に供給される情報量の少ない画像に対しては量子化ステップをあまり小さくし過ぎず、その後に続く情報量が多い画像が符号化されるまで出力バッファの残量を低く保つべきであるのに、前述した出力バッファ残量をフィードバックする方式では、上記情報量が少ない画像が連続するうちに出力バッファの残量を増加させてしまうようになる。

【0041】逆に、情報量が多い画像の後に情報量の少ない画像が続く場合では、先に供給される情報量の多い画像を大きな量子化ステップで圧縮して出力バッファの残量を減らさなくても、その後に続くのは情報量の少ない画像なのでオーバーフローし難いはずであるが、上記出力バッファ残量フィードバック方式では、続く画像の情報量がわからないためバッファの残量を減らす方向、すなわち量子化ステップを大きくする方向に制御し、画質を低下させてしまう。

【0042】さらに、入力画像の情報量の急激な変化の顕著な例がシーンチェンジであるが、シーンチェンジは前の画像との相関が極めて低いため、フレーム内符号化で圧縮したほうが効率良く圧縮できるが、予め情報量を知っており、かつ前後の画像のそれとの比較ができなければシーンチェンジかどうか、すなわち、フレーム内符号化で圧縮するか、あるいはフレーム間予測符号化で圧縮するか判断することができない。

【0043】またさらに、フレーム内符号化で圧縮するには出力バッファに十分な空き容量が必要であるが、前の画像の画質を著しく損なうことなく出力バッファに十分な空き容量を確保するためには、充分前のフレームから準備を行い、当該出力バッファの残量を減らしておかなければならない。したがって、フレーム内符号化で圧縮することは圧縮を行うよりも充分以前に判っていなければならない。

【0044】そこで、本発明は、上述のような実情に鑑みて提案されたものであり、シーンチェンジが検出で

(7)

11

き、効率の良い画像圧縮が可能で、全体的に画質を向上させることができると共に、コストの低減も可能な画像符号化装置を提供することを目的とするものである。

【0045】

【課題を解決するための手段】本発明の画像符号化装置は、上述した目的を達成するために提案されたものであり、入力画像データを複数枚蓄える画像データ蓄積手段と、上記画像データ蓄積手段に蓄積された入力画像データを複数のブロックに分割し、それぞれのブロック毎にサーチ側とリファレンス側のピクセル毎に求めた誤差値を1画面分合計した第1の合計値と、時間的に過去の1画面分の誤差値を合計した第2の合計値との比に基づいてシーンチェンジ検出を行うシーンチェンジ検出手段と、上記シーンチェンジ検出手段によるシーンチェンジ検出出力に応じて画像データの圧縮方法を選択する圧縮方法選択手段と、上記圧縮方法選択手段により選択された圧縮方法に従って上記入力画像データに所定の圧縮符号化処理を施す圧縮符号化処理手段と、圧縮符号化された画像データを蓄積する出力バッファ手段と、上記シーンチェンジ検出手段によるシーンチェンジ検出出力に応じて上記出力バッファ手段の蓄積量を制御するバッファ制御手段とを有することを特徴とするものである。

【0046】ここで、上記シーンチェンジ検出手段は、上記第1の合計値が所定の閾値以下であるときには上記第2の合計値を保存しておく（更新しない）こと、及び／又は、シーンチェンジを検出したときに上記第2の合計値を保存しておく（更新しない）ことを行う。また、本発明の画像符号化装置は、入力画像データを複数枚蓄える画像データ蓄積手段と、上記画像データ蓄積手段に蓄積された入力画像データを複数のブロックに分割し、それぞれのブロック毎にサーチ側とリファレンス側のピクセル毎に求めた差分値の第1の絶対値和と、時間的に過去の差分値の第2の絶対値和との比に基づいてシーンチェンジ検出を行う第1のシーンチェンジ検出手段と、上記第1のシーンチェンジ検出手段でシーンチェンジと判定したブロックの1画面に対する割合に基づいて、当該1画面のシーンチェンジを検出する第2のシーンチェンジ検出手段と、上記第2のシーンチェンジ検出手段によるシーンチェンジ検出出力に応じて1画面の画像データの圧縮方法を選択する圧縮方法選択手段と、上記圧縮方法選択手段により選択された圧縮方法に従って上記入力画像データに所定の圧縮符号化処理を施す圧縮符号化処理手段と、圧縮符号化された画像データを蓄積する出力バッファ手段と、上記シーンチェンジ検出手段によるシーンチェンジ検出出力に応じて上記出力バッファ手段の蓄積量を制御するバッファ制御手段とを有することを特徴とするものでもある。

【0047】このときも、上記第1のシーンチェンジ検出手段は、上記第1の絶対値和が所定の閾値以下であるときには上記第2の絶対値和を保存しておく（更新しない）こと、及び／又は、シーンチェンジを検出したときに上記第2の絶対値和を保存しておく（更新しない）ようにする。

12

い）こと、及び／又は、シーンチェンジを検出したときに上記第2の絶対値和を保存しておく（更新しない）ようにする。

【0048】

【作用】本発明によれば、直前の誤差値（第1の誤差値）と過去の誤差値（第2の誤差値）とを比較することで、ブロック毎にシーンチェンジであるか否かを検出している。また、本発明によれば、直前の差分の絶対値和（第1の絶対値和）と過去の差分の絶対値和（第2の絶対値和）とを比較してブロック毎にシーンチェンジを検出し、さらにシーンチェンジと判定されたブロックの1画面中の割合からこの1画面がシーンチェンジであるか否かを検出している。

【0049】ここで、第1の誤差値や第1の差分の絶対値和が所定の閾値以下のときには、過去の第2の誤差値や第2の差分の絶対値和を保存しておく（更新しない）ことで、誤差値や差分の絶対値和の変動が少ない画像のシーンチェンジの検出精度を上げている。また、第1の誤差値や第1の差分の絶対値和からそのブロックがシーンチェンジであると検出したときにも、過去の第2の誤差値や第2の差分の絶対値和を保存しておく（更新しない）ことで、次のブロックのシーンチェンジの検出を可能としている。

【0050】

【実施例】以下、図面を参照し、本発明の実施例について詳述する。図1には本発明実施例の画像符号化装置の概略構成を示す。なお、この図1において、前述した図9と同じの構成については同一の指示符号を付してその説明については省略する。

【0051】この図1の構成において、前記図9の構成に追加された構成要素はシーンチェンジ検出回路31と圧縮方法選択回路32と動きベクトル発生回路34であり、また、フレームメモリ40と動き検出器41及び量子化ステップ制御器33が変更されている。すなわち、本発明の第1の実施例の画像符号化装置は、入力画像データを複数枚フレーム分蓄える画像データ蓄積手段であるフレームメモリ40と、上記フレームメモリ40に蓄積された入力画像データを複数のブロックに分割し、それぞれのブロック毎にサーチ側とリファレンス側のピクセル毎に求めた誤差値を1画面分合計した第1の合計値（SAD）と、時間的に過去の1画面分の誤差値を合計した第2の合計値（prev __SAD）との比に基づいてシーンチェンジ検出を行うシーンチェンジ検出回路31と、上記シーンチェンジ検出回路31によるシーンチェンジ検出出力に応じて画像データの圧縮方法（フレーム内符号化／フレーム間予測符号化）を選択する圧縮方法選択回路32と、上記圧縮方法選択回路32により選択された圧縮方法に従って上記入力画像データに所定の圧縮符号化処理を施す圧縮符号化処理手段である差分器12から可変長符号化回路17まで及びローカルデコーダと、圧縮符

(8)

13

号化された画像データを蓄積する出力バッファ18と、上記シーンチェンジ検出回路32によるシーンチェンジ検出出力に応じて上記出力バッファ18の蓄積量を制御するバッファ制御手段としての作用を有する量子化ステップ制御器33とを有することを特徴とするものである。

【0052】また、本発明の第2の実施例の画像符号化装置は、上記図1のシーンチェンジ検出回路31において、上記フレームメモリ40に蓄積された入力画像データを複数のブロックに分割し、それぞれのブロック毎にサーチ側とリファレンス側のピクセル毎に求めた差分値（誤差）の第1の絶対値和(AD[i])と、時間的に過去の差分値（誤差）の第2の絶対値和(prev AD[i])との比に基づいてシーンチェンジ検出を行う（第1のシーンチェンジ検出手段）と共に、シーンチェンジと判定したブロックの1画面に対する割合に基づいて、当該1画面のシーンチェンジを検出する（第2のシーンチェンジ検出手段）ようにしている。

【0053】この図1において、まず、入力端子1から入力された入力画像データは、フレームメモリ40に蓄えられる。このフレームメモリ40は、図9のフレームメモリ10とは異なり、所定数のフレームを蓄積できるものである。このときの蓄積する所定数としては、多過ぎるとフレームメモリ40が大規模になってしまうので好ましくない。上記所定数として効率的な長さ（フレーム数）は、ビットレートと出力バッファ18の容量、フレーム内符号化の圧縮方式の画像同士の間隔（ほとんどの場合GOPの長さといっても差し支えない）に大きく依存する。これは圧縮方法及び圧縮率の違いから生ずる圧縮データの大きさのむらを上記出力バッファ18によって吸収し、定ビットレートにすることができ範囲が、上記ビットレート及び出力バッファ容量とフレーム内符号化がなされる画像同士の間隔等の条件によって制約されるからである。

【0054】ところで、一般的にフレーム内符号化方式で圧縮することは定期的に行われる（これがGOPの区切りになることが多い）ものであり、このフレーム内符号化の圧縮方式は当該圧縮後のデータ量が他の方式（フレーム間予測符号化）に比べてかなり大きいものである。このため、当該フレーム内符号化による圧縮画像同士（或いはGOP）の間隔で情報量を調べ、データ量の配分をするのは、一つの合理的な方法である。

【0055】しかし、本実施例の方式では、後述するシーンチェンジ等によって前後の画像の相関が著しく低くなった場合にも、フレーム内符号化方式で圧縮するようにしている。その場合、定期的に行うはずである近傍のフレーム内符号化によって圧縮を行う画像は、定ビットレート或いは均質な画質の維持が困難になるため、フレーム内符号化による圧縮である必然性を失い、フレーム内符号化で圧縮することを取り止める必要がでてくる。

14

【0056】したがって、フレーム内符号化による圧縮を行う周期の2倍程度を上記フレームメモリ40の長さ（上記所定数）とすることが適当である。もちろん、上記フレーム数は一例であり、これに限定されることなく様々な条件に合わせて変更することは可能である。次に、シーンチェンジ検出回路31は、入力画像のシーンチェンジを後述するように動き検出で得たパラメータを用いて検出する。

【0057】ここで、当該シーンチェンジ検出回路31においてシーンチェンジを検出する目的は、フレーム間予測符号化かフレーム内符号化のいずれかの圧縮方式を決定するための判断材料にすることが主である。それは、前後で相関の極めて低い画像では、フレーム間予測符号化で圧縮するよりもフレーム内符号化で圧縮する方が効率良く圧縮できるからである。また、圧縮後のデータも大きなものとなるため、データ量配分や出力バッファマネジメントの観点からもシーンチェンジを把握することは重要である。

【0058】また、上記フレームメモリ40に蓄積された画像データは、適宜、画像情報評価回路50にも送られる。ここで、当該画像情報評価回路50は、大別して2通りのパラメータを算出するものである。第1のパラメータは、フレーム内符号化で圧縮を行った場合の圧縮後のデータ量を予測することが可能なように、その画像自身の情報量を示すものである。この第1のパラメータとしては、例えば、フレームメモリ40から供給された画像データに対して、DCT処理をブロック毎に行い、そのDCT係数の和や統計をとったものとしたり、また、それでは規模が大きくなる場合には、平均自乗誤差のブロック毎の和を求めたものとする。いずれにしても、当該画像情報評価回路50では、画像の情報量を表し、圧縮後のデータ量を類推するに足るパラメータを算出する。第2のパラメータは、フレーム間予測符号化で圧縮を行った場合の圧縮後のデータ量を予測することが可能な、画像の差分情報量を示すものである。この場合のパラメータとしては、例えば、フレームメモリ40に格納された画像と動き補償後の画像との差分値のブロック内の和を用いる。このパラメータ算出の際には、一般的な動きベクトル検出回路で得られる動きベクトルが検出された最小誤差を利用することができる。

【0059】次に、圧縮方法選択回路32については説明する。当該圧縮方法選択回路32は、上記シーンチェンジ検出回路31からの出力と画像情報評価回路50からの情報量パラメータとに基づいて、フレーム内符号化／フレーム間予測符号化（P、Bピクチャ）のいずれの圧縮方式で圧縮を行うのかを選択する回路である。ここで、フレーム内符号化方式による圧縮では少なくともGOPの最初になければならない。GOPはランダムアクセスを考慮してある程度の間隔とするので、必然的にIピクチャはある程度定期的に発生するし、シーンチェン

(9)

15

ジ等によっても発生する。したがって、画像のカウントとシーンチェンジ検出回路 31 の出力が当該圧縮方法選択回路 32 に加えられるようになっている。

【0060】次に本実施例の構成における処理の流れを、図 2 のフローチャートに沿って説明する。まず、ステップ S1 では、入力端子 1 に入力された画像データが順次フレームメモリ 40 へ格納される。ここで、先に述べたように I ピクチャの頻度や間隔の決定が画質に影響を及ぼすので、これに関係して符号化に先だって GOP を決めておく必要があり、また、後述するようにレートコントロールをするために符号化に先だって 1 GOP 分の画像についての情報を収集しなければならない。このように、次々と入力されてくる画像データに対してその間の分析をし、符号化するまでの十分な遅延時間を稼ぐため、大量のフレームメモリ 40 を用いる。

【0061】次に、ステップ S2 では、フレーム内符号化で圧縮するために必要な動きベクトルを動き検出器 41 及び動きベクトル発生回路 34 で検出する。すなわち、このステップ S2 では、上記フレームメモリ 40 に格納された画像データは予め定められたスケジュールでメモリ中の各画像データを P ピクチャ或いは B ピクチャ *

16

* として圧縮符号化できるように、先ず動き検出器 41 で動き検出（モーションエスティメーション）を行う。なお、動き検出を行う画像については I ピクチャを規定しない。それはどの画像データが I ピクチャになるのかこの時点では確定していないからであり、また I ピクチャは動き補償を必要としないため、後にどの画像データでも I ピクチャにすることが可能だからである。

【0062】ここでは、動き検出をする際に用いられる最小歪み（Minimum Distortion）或いは誤差（差分値）の絶対値和（AD：Absolute Difference）と呼ばれるものを符号化に用いるパラメータの一つとして読み出し格納する。なお、上記誤差の絶対値和（AD）は、リファレンス側の画像を 8×8 ピクセルのブロックに分割し、輝度データ $8 \times 8 \times 4$ ピクセルと色差データ $8 \times 8 \times 2$ ピクセルのマクロブロック（MB）に対し、動き検出を行なった際に求めた動きベクトルで切りだされるサーチ側のマクロブロックとの各画素同士の差の絶対値和で、以下の式(1)で求めることができる。

【0063】

【数 1】

$$blockAD = \sum_{i=1}^8 \sum_{j=1}^8 (abs(ref[i, j] - search[i, j])) \quad (1)$$

$ref[i, j]$: リファレンス MB 位置 (i, j) の画素

$search[i, j]$: サーチ MB 位置 (i, j) の画素

【0064】これをさらにマクロブロック内のブロックについて集計したものをを用いてマクロブロックの誤差の絶対値和（AD）とする。このパラメータはシーンチェンジの判定やフレーム間予測符号化で圧縮する場合の画像の相関も考慮した情報量の推定に用いる。このパラメータはまた、マクロブロックタイプを決定するために使われる。

【0065】画像の情報量を推定するパラメータ（SAD）は、式(2)のように、一つの画像内の上記誤差の絶対値和（AD）を合計したものである。

$$SAD = \sum AD \quad (2)$$

もちろん、上記誤差の絶対値和（AD）以外に最小歪み（Minimum Distortion）を用いてもよい。

【0066】次に、ステップ S3 では、上記シーンチェンジ検出回路 31 でシーンチェンジの検出を行う。このシーンチェンジ検出回路 31 でのシーンチェンジ検出は、動き検出器 41 で得られた上記パラメータ AD を使って、シーンチェンジを検出する。ここで、シーンチェンジ検出回路 31 におけるシーンチェンジ検出のロジックには 2 つのバリエーションがある。

【0067】すなわち、第 1 の実施例に対応する第 1 のシーンチェンジ検出方法は、上記誤差の絶対値和（AD）を 1 画面分合計した合計値（SAD）を使うもので

あり、第 2 の実施例に対応する第 2 のシーンチェンジ検出方法は、上記誤差の絶対値和（AD）を使い各マクロブロックで判定を行い、シーンチェンジと判定された割合で検出する。これらどちらも、考え方は、過去の誤差の絶対値和（AD）の値から検出しようとしている現在の誤差の絶対値和（AD）の値が、ある比率以上に大きくなった場合をシーンチェンジとするものである。

【0068】すなわち、この第 1 のシーンチェンジ検出方法では、例えば、あるフレームから前方向の動きベクトルの検出を行ったときにマクロブロック毎の誤差の絶対値和を 1 画面分（1 フレーム分）合計した値（合計値（SAD））と、過去のフレームで同様に前方向の動きベクトル検出を行ったときにマクロブロック毎の誤差の絶対値和を 1 画面分（1 フレーム分）合計した値（合計値（prev_SAD））との比（SAD / prev_SAD）が、ある閾値（Thresh）を越えた場合は当該フレームがシーンチェンジであると判定する。

【0069】ただし、この第 1 のシーンチェンジ検出方法の場合、誤差の絶対値の合計値（SAD）が、ある閾値（min_SAD）と比べて異常に小さい場合には、直前の誤差の絶対値の上記合計値（prev_SAD）を更新しないようにする。また、シーンチェンジであると判定された場合は、直前の上記合計値（prev_SAD）を更新しないようにす

(10)

17

る。

【0070】上記第1のシーンチェンジ検出方法の詳細なフローチャートは、図3に示すようになる。この図3において、ステップS201では初期値の設定を行う。この初期値設定はシーンチェンジが始まる前に行う。この初期値として、変数(small_SAD)は、通常の連続したシーンでの少なめの前記合計値(SAD)とし、適切な値に設定する。また、変数(thresh)は、例えば3を設定しておく。

【0071】次のステップS202では各フレーム毎に地点Aを設定し、ステップS203では動き検出を行い、前記誤差の絶対値和(AD)を1画面分集計して合計値(SAD)を求める。次に、ステップS204では、前述したように、あるフレームにおける合計値(SAD)と、過去のフレームにおける合計値(prev_SAD)との比($SAD / prev_SAD$)が、ある閾値(Thresh)を越えたか否かの判定を行う。当該ステップS204において、越えたと判定した場合にはステップS206でシーンチェンジを検出したと判定し、その後ステップS209に進んで地点Bへ抜ける。一方、ステップS204において、越えないと判定した場合にはステップS205でシーンチェンジでないと判定する。

【0072】ステップS205の次はステップS207に進む。このステップS207では、誤差の絶対値の合計値(SAD)が、ある閾値(min_SAD)と比べて小さいか否かの判断を行う。このステップS207において、小さくないと判定した場合にはステップS209に進んで地点Bへ抜ける。一方、ステップS207において、小さいと判定した場合には、ステップS208に進む。

【0073】ステップS208では、過去のフレームの合計値(prev_SAD)を当該フレームの合計値(SAD)とすることで、合計値を更新しないようにし、その後、ステップS209に進んで地点Bへ抜ける。次に、上記誤差の絶対値和(AD)を使用して、図4に示すような縦15マクロブロック×横22マクロブロックからなる画像の各マクロブロック毎の比較判定によって、シーンチェンジを検出する第2のシーンチェンジ検出方法について説明する。

【0074】この第2のシーンチェンジ検出方法では、例えば、あるフレームから前方向の動きベクトルの検出を行ったときに、マクロブロック毎の誤差の絶対値和($AD[i]$)と、過去(直前)のフレームで同様に前方向の動きベクトル検出を行ったときにマクロブロック毎の誤差の絶対値和($prev_AD[i]$)との比($AD[i] / prev_AD[i]$)が、ある閾値(Thresh)を越えた場合は、そのマクロブロックはシーンチェンジとなっていると判定する。

【0075】ただし、この第2のシーンチェンジ検出方法の場合、マクロブロック毎の誤差の絶対値和($AD[i]$)が、ある閾値(min_AD)と比べて異常に小さい場合は、マクロブロック毎の直前の誤差の絶対値和($prev_AD$

18

[i])を更新しない。また、シーンチェンジであると判定されたマクロブロックでは、マクロブロック毎の直前の誤差の絶対値和($prev_AD[i]$)を更新しない。

【0076】さらに、この第2のシーンチェンジ検出方法では、上述のようにしてシーンチェンジと判定されたマクロブロックの数(MB_SC_num)と、1画面(1フレーム)のマクロブロックの数の合計との比($MB_SC_num / 1画面のマクロブロック数$)が、ある閾値(thresh)を越えた場合は、そのフレームはシーンチェンジであると判定する。

【0077】このように、第2のシーンチェンジ検出方法では、第1のシーンチェンジ検出方法のような単純な検出と異なり、マクロブロック毎に判定を行なうため、それぞれのマクロブロック領域でのシーンチェンジ判定ができ、より精度の高いシーンチェンジ検出が可能となる。また、シーンチェンジと判定されたマクロブロックでは、マクロブロックタイプの判定でフレーム内符号化のマクロブロック(イントラマクロブロック)とする等の副次的な利用も有効である。

【0078】したがって、第2のシーンチェンジ検出方法では、第1のシーンチェンジ検出方法に比べて、例えばアニメーションのような画像の場合のシーンチェンジの誤検出低減に役立つ。なお、当該アニメーション画像のシーンチェンジ検出については後述する。第2のシーンチェンジ検出方法の詳細なフローチャートは、図5に示すようになる。

【0079】この図5において、ステップS211では初期値の設定を行う。この初期値設定はシーンチェンジが始まる前に行う。この初期値として、変数(small_AD)は、通常の連続したシーンでの少なめの前記誤差の絶対値和(AD)とし、また、変数(thresh)は、例えば0.3を設定しておく。次のステップS212では各フレーム毎に地点Aを設定し、ステップS213では動き検出を行う。

【0080】次に、ステップS214では、マクロブロックの数を示す変数(MB_SC_num)とフレーム内のマクロブロックの番号(i)をそれぞれ初期化する。その後、マクロブロック毎の判定の処理に移る。このマクロブロック毎の判定の処理では、まず、ステップS215において、前述したように、あるフレームのマクロブロック毎の誤差の絶対値和($AD[i]$)と、過去(直前)のフレームにおけるマクロブロック毎の誤差の絶対値和($prev_AD[i]$)との比($AD[i] / prev_AD[i]$)が、ある閾値(Thresh)を越えたか否かの判定を行う。当該ステップS215において、越えたと判定した場合にはステップS217でシーンチェンジを検出したと判定し、その後ステップS219に進む。一方、ステップS215において、越えないと判定した場合にはステップS216でシーンチェンジでないと判定した後、ステップS218に進む。

(11)

19

【0081】上記ステップS219ではマクロブロックの数を示す変数(MB_SC_num)を1ずつ増加させ、ステップS218では過去(直前)のフレームにおけるマクロブロックの誤差の絶対値和(prev_AD[i])当該フレームのマクロブロックの誤差の絶対値和(AD[i])とすることで、合計値を更新しないようにする。これらステップS218とS219の後はステップS220に進み、ここでマクロブロックの番号(i)を1ずつ増加させる。

【0082】次のステップS221では、マクロブロックの番号(i)が1画面(1フレーム)のマクロブロック数以上となったか否かの判定を行い、なっていないと判定した場合にはステップS215に戻り、なっていると判定した場合にはステップS222に進む。ステップS222では、マクロブロックの数(MB_SC_num)と、1画面(1フレーム)のマクロブロックの数の合計との比(MB_SC_num / 1画面のマクロブロック数)が、ある閾値(thresh)を越えたか否かの判定を行う。このステップS222において、越えたと判定した場合にはステップS223において当該フレームはシーンチェンジであると判定し、越えていないと判定した場合にはステップS224で当該フレームはシーンチェンジでないと判定する。

【0083】図2のフローチャートに戻って、ステップS3の上述したシーンチェンジ検出の後のステップS4では、GOP長の決定を行い、ステップS5では圧縮方法の選択(ピクチャタイプの決定)とを行う。すでに述べたように、符号化に際してランダムアクセス性を考慮して適当なフレーム数毎にGOPを区切る。このとき少*

20

*なくともGOPの符号順で最初のピクチャはIピクチャでなければならないからピクチャの数をカウントし定期的にピクチャタイプをIピクチャにする。

【0084】一方、上記シーンチェンジによって前後のピクチャで相関が低くなった場合、これも先に述べたようにIピクチャで圧縮符号化すると効率が良い。しかしながら、Iピクチャは圧縮率が低いいため、低ビットレートにおいては頻繁に現れると画質の低下を招く。したがってシーンチェンジが検出された場合、Iピクチャ同士の間隔を適度に保つよう適応的にGOPの長さを決める。

【0085】次のステップS6では、マクロブロックタイプの判定を行う。すなわち、このステップS6では、マクロブロック毎の圧縮方法と、マクロブロックタイプとを決める。ここで、マクロブロックタイプの決定のためには、先ず画像情報評価回路50において、先ず、上記動き検出で得られたパラメータ以外に誤差の平均絶対値和(MAD: Mean Absolute Difference)、アクティビティ(Activity)を各画像毎に評価する。

【0086】上記誤差の平均絶対値和(MAD)はIピクチャの情報量を推定するためのパラメータで、下記の式(3)によって8×8画素のブロック毎に求める。また、必要に応じてマクロブロック或いは画面で集計を行う。このパラメータはまた、マクロブロックタイプを決定するために使われる。

【0087】

【数2】

$$blockMAD = \sum_{i=1}^8 \sum_{j=1}^8 (X(i, j) - \bar{X}) \quad (3)$$

$X(i, j)$: 位置(i,j)の画素

\bar{X} : ブロック内画素の平均値

【0088】これをさらに、式(4)のように、マクロブロック内のブロックについて合計したものを用いてマクロブロックの判定に用いる。

$$MAD = \sum blockMAD \quad (4)$$

またさらに、式(5)のように、マクロブロックの値を一つの画像内で合計してその値を、その画像の(Iピクチャとしての)情報量を表すパラメータSMADとする。

$$SMAD = \sum MAD \quad (5)$$

また、上記アクティビティは、一つの画面の中でそのマクロブロックの画像の状態に応じて、よりきめ細かに量子化ステップを制御することにより画質を維持しながら、より圧縮効率を高めるためにその画像の状態を定量化するパラメータである。

【0090】例えば一つのブロック内で画像が画素のレベル変化の少ない平坦な部分(フラットな部分)では量子化による歪みが目立ち易く、量子化ステップを小さく

してやるべきで、逆にレベル変化が多い複雑なパターンのブロックでは量子化歪みは目立ち難く、情報量も多いので量子化ステップを大きくするべきである。そこで、例えばブロックの平坦度(フラットネス)を表すパラメータをこのアクティビティとして用いるようにしている。

【0091】上述のようにして求めた、上記誤差の平均絶対値和(MAD)と既に求めている誤差の絶対値和(AD)は、それぞれフレーム内符号化/フレーム間予測符号化で圧縮したときの圧縮後のデータ量に関係するので、この2つのパラメータを比較すればフレーム内符号化/フレーム間予測符号化のどちらのマクロブロックタイプがより少ないデータ量になるか判定できる。

【0092】次のステップS7では、レートコントロールのためのビット配分を行う。各ピクチャ毎の圧縮符号化された後のデータサイズは、その符号化方式や元々の

(12)

21

画像データが持つ情報量、前後の相関などによって大きく変動する。平均的な画質を保つようにするならばことさらである。各ピクチャ毎の圧縮符号化された後のデータサイズのむらは出力バッファ18によってある程度吸収されるが、平均的には一定のビットレートにしなければならない。したがって、ある区間を定めればその間のピクチャのトータルの圧縮後のデータ量が決まる。そこで、既に決定しているピクチャタイプと、予め調べておいた画像の情報量パラメータとを用いて各ピクチャ毎に*

$$\text{Total Bit Count} = (\text{Bit Rate} [\text{bit/s}] \times \text{Number Of Picture In GOP} [\text{picture}]) / (\text{Picture Rate} [\text{picture/s}]) \quad [\text{bit}] \quad (6)$$

$$\text{Available Bits} = (\text{Total Bit Count} \times \text{ターゲットの画像情報量パラメータ}) / \text{画像情報量パラメータのGOP合計値} \quad [\text{bit}] \quad (7)$$

ここでいう情報量パラメータは、先に述べたパラメータSMAD、SADを用い、これに圧縮するピクチャタイプ別の乗数をかけたものである。また、上記乗数は各ピクチャタイプ間のパラメータと画質との関係を調整する※

*圧縮後のデータ量、すなわち各ピクチャが使って良いビットの量を定める。

【0093】情報量の少ない画像やBピクチャには少なく、情報量の多い画像やIピクチャには多くする。これをビット配分と呼ぶ。これによって画質のばらつきを抑え、なおかつ一定レートに保つことが容易になる。例えば本実施例ではGOPをその区間として、次の式(6)、式(7)のようにビット配分を行う。

※ものである。

【0094】なお、上記式(7)の上記画像情報量パラメータのGOP合計値は、式(8)に示すようにして求める。

$$\text{画像情報量パラメータのGOP合計値} = K_i \times \Sigma \text{SMAD} + K_p \times \Sigma \text{SADp} + K_b \times \Sigma \text{SADb}$$

SADp: PピクチャのSAD

SADb: BピクチャのSAD

(8)

次のステップS8では、量子化ステップ制御器33において基本量子化ステップの決定及び画面内の量子化ステップの制御を行う。

【0095】上述のようにしてピクチャタイプが決まり、マクロブロックタイプが決まれば、上記画像情報評価回路50によって求めた情報量パラメータと量子化ステップとから量子化後のデータサイズをある程度予測できる。したがって、情報量パラメータと量子化後のデータ量が決まっていれば、量子化ステップを逆算することができる。このようにして求めた量子化ステップを、そのピクチャの基本量子化ステップとする。

【0096】量子化ステップ制御器33では、上述のように画面内の量子化ステップを各ブロック毎になるべく画質を高く、かつ、圧縮効率も高くするように制御する。この制御は、上記アクティビティやマクロブロックタイプなどの情報を基に基本量子化ステップを制御することで実現する。次のステップS19では、符号化を行う。

【0097】上述のようにして圧縮符号化の全てのパラメータが決まっているので、その後はMPEGの規則にしたがって圧縮符号化する。最後に、ステップS11では、予測パラメータの更新を行う。ここで、画像情報量と基本量子化ステップ、圧縮後のデータ量の関係は、圧縮する画像に依存する。したがって、ここでは、その関係を表す式に用いるパラメータ、予測パラメータを、圧縮後の実際のデータ量をフィードバックすることにより学習させ、予測の精度を向上させている。

【0098】次に、前述したアニメーション画像のシーンチェンジ検出について、以下に説明する。アニメーション画像についての前記画像の情報量を推定するパラメ

ータである誤差の絶対値和(AD)を1画面分合計した合計値(SAD)は、図6、図7に示すように、動きのある部分でも、実際には0に近い値と、より大きな値の繰り返しになっている。これはアニメーションの多くの場合、画像(絵)の変化は秒当たり30フレームに対して、毎フレームではなく、例えば7フレーム/秒等になっているためである(さらに少ない場合もある)。なお、比較のため、図8には、アニメーション画像ではなく実写画像の場合の合計値(SAD)を示している。

【0099】このように、コマ数の少ないアニメーションでは、少ないコマ数で動きをつけるために、動きのある部分は上記合計値(SAD)が比較的大きな値をとることがある。図7に示すように、クロスフェード部を除いた上記合計値(SAD)の最大値は2863になっている。一方、図6のシーンチェンジ部での合計値(SAD)は、8366になっている。ここでもしも、この2つの部分が連続していた場合、第1のシーンチェンジ検出方法を行うと検出できないことになる。

【0100】そこで、第2のシーンチェンジ検出方法において、アニメーションのような画像のシーンチェンジを行う場合には、以下のようにする。まず、マクロブロック毎に合計値(SAD)の変化率を求める。次に、合計値(SAD)は、検索画像と参照画像の距離が遠くなると大きくなる傾向にあるので、距離の影響を小さくするため、式(9)に示すように、補正係数 K_{MOD} で割る。

【0101】

$$\text{補正されたSAD} = \text{SAD} / K_{MOD} \quad (9)$$

ここで、式(10)に示すように、 K_{MOD} は距離が遠くなるに従って大きくなる。 $K_{MOD} = 1 + 0.5 \times ((\text{検索画像と参照画像の距離}) - 1)$ (10)次に、式(11)に

50

(13)

23

示すように、マクロブロック毎の合計値（SAD）の変化率を求める。

$$\text{SAD変化率} = (\text{MBのSAD}) / (\text{MBの前のSAD}) \quad (11)$$

また、このSAD変化率が決められた閾値 T_{SAD} を越えるマクロブロック（MB）の数を C_{OVER} とする。例外処理として、あるマクロブロックについて式(11)の計算時に、もしも分母（マクロブロックの前のSAD）が0で、分子（マクロブロックのSAD）が0でない場合、※

$$T_{\text{MB}} = (\text{画面全体のマクロブロック数}) \times (\text{判定比率}) \quad (12)$$

これは、例えば、判定比率を0.5にすれば、画面全体のうち半分以上の面積が大きく変わったらシーンチェンジとすることを近似していることになる。

【0104】上述のようなことから、このシーンチェンジ判定方法では、大きく変化のあった部分の面積が画面全体に対してどれくらいの比率を占めているかをシーンチェンジの判定規準にしている。これにより、画面の小さな部分だけに大きな変化が起きた場合でも、シーンチェンジと判定してしまうことを防ぐことができる。また、画面の面積でどのくらい大きく変わったらシーンチェンジとみなすかを閾値を変えることで直接的に制御できるようにする。

【0105】上述したように、本発明実施例の画像符号化装置においては、圧縮符号化の構成の前段に数フレーム分のフレームメモリ40と、入力画像データの情報量を評価する画像情報評価回路50を有し、さらにシーンチェンジ検出回路31によって画像間の相関等を調べてシーンチェンジを検出するようにしている。本実施例装置では、上記シーンチェンジが検出されたフレームをフレーム内符号化で圧縮できるように、充分前のフレームから量子化ステップ制御器33で量子化器15の量子化ステップを制御し、出力バッファ18の残量を減らしておけるようにしている。このため、シーンチェンジ検出回路31と圧縮符号化処理手段との間に充分な容量のフレームメモリ40を備えている。

【0106】このように、本発明実施例装置においては、画像情報評価回路50によって符号化に先だって入力画像データが本来持っている情報量の大小を数フレーム先まで事前に見積もり、それに合わせてビット量をそれぞれの画像に配分し、そのとき、配分したビット量の総和から導かれるビットレートが一定になるようにする。また、前の画像と相関が著しく低くなる場合、すなわちシーンチェンジを検出したときには、圧縮方法をフレーム内符号化に変更している。さらに、本実施例装置では、圧縮方法に応じたデータ量の配分をとることが可能なため、シーンチェンジが検出されるような前の画像と相関が著しく低くなる画像でも画質の劣化を抑えて圧縮することができる。

【0107】

【発明の効果】本発明においては、直前の誤差値（第1の誤差値）と過去の誤差値（第2の誤差値）とを比較す

24

*【0102】

*

※閾値 T_{SAD} を越えるものとして処理する。

【0103】最後に、もしも式(12)に示すような場合において、 $T_{\text{MB}} < C_{\text{OVER}}$ なら、シーンチェンジという判定をする。

10 ることで、ブロック毎にシーンチェンジであるか否かを検出することができ、効率の良い画像圧縮が可能となり、全体的に画質が向上する。また、シーンチェンジ検出の為に新たに追加した回路は非常に簡単であり、コストがかからず、さらに回路は非常に簡単である割にはシーンチェンジの検出精度が高く、効果が大きい。さらに、もしも人間には認識できるがこのシーンチェンジ検出方法で検出できないシーンチェンジがあったとしても、それは変動率が小さい場合なので、その画像に特別に多くのビットを配分する必要はなく、したがってバッファ残量を空けておく必要もないため、未検出が画質劣化の問題になり難い。

【0108】ここで、第1の誤差値が所定の閾値以下のときには、過去の第2の誤差値を保存しておく（更新しない）ことで、誤差値の変動が少ない画像のシーンチェンジの検出精度を上げることができ、したがって、効率の良い画像圧縮が可能となる。また、第1の誤差値からそのブロックがシーンチェンジであると検出したときにも、過去の第2の誤差値を保存しておく（更新しない）ことで、引き続き来るシーンチェンジを検出でき、効率の良い画像圧縮が可能となる。さらに、これらのことを同時に行えば、さらに効率の良い画像圧縮が可能となる。

【0109】また、本発明においては、直前の差分の絶対値和（第1の絶対値和）と過去の差分の絶対値和（第2の絶対値和）とを比較してブロック毎にシーンチェンジを検出し、さらにシーンチェンジと判定されたブロックの1画面中の割合からこの1画面がシーンチェンジであるか否かを検出しているため、シーンチェンジの検出率が向上し、また、画面内での変更箇所及び比率が前もってわかるため、ビット配分やマクロブロックタイプの判定にも利用でき、より効率の良い画像圧縮が可能となり、全体的に画質が向上する。

【0110】この場合も、第1の差分の絶対値和が所定の閾値以下のときには、過去の差分の絶対値和を保存しておく（更新しない）ことで、差分の絶対値和の変動が少ない画像のシーンチェンジの検出精度を上げることができ、したがって、効率の良い画像圧縮が可能となる。また、第1の差分の絶対値和からそのブロックがシーンチェンジであると検出したときにも、過去の第2の差分の絶対値和を保存しておく（更新しない）ことで、引き

(14)

25

続き来るシーンチェンジを検出でき、効率の良い画像圧縮が可能となる。さらに、これらのことを同時に行えば、さらに効率の良い画像圧縮が可能となる。

【図面の簡単な説明】

【図1】本発明実施例の画像符号化装置の概略構成を示すブロック回路図である。

【図2】本実施例装置の動作を説明するためのフローチャートである。

【図3】誤差値の合計値（SAD）を使用したシーンチェンジ検出（第1のシーンチェンジ検出方法）のフローチャートである。

【図4】マクロブロック毎のシーンチェンジ判定の説明に使用する図である。

【図5】差分値（誤差）の絶対値和（AD）を使用したシーンチェンジ検出（第2のシーンチェンジ検出方法）のフローチャートである。

【図6】画像の例としてアニメーション画像にシーンチェンジが含まれるときの誤差値の合計値（SAD）を示す図である。

【図7】画像の例としてアニメーション画像にクロスフェードが含まれるときの誤差値の合計値（SAD）を示す図である。

【図8】画像の例として実写画像の誤差値の合計値（SAD）を示す図である。

【図9】従来の画像符号化装置の概略構成を示すブロック回路図である。

【図10】画像の解像度と構成について説明するための図である。

【図11】マクロブロックとブロックについて説明する

26

ための図である。

【図12】ルミナンスピクセルとクロミナンスピクセルの配置について説明するための図である。

【図13】ジグザグスキャンについて説明するための図である。

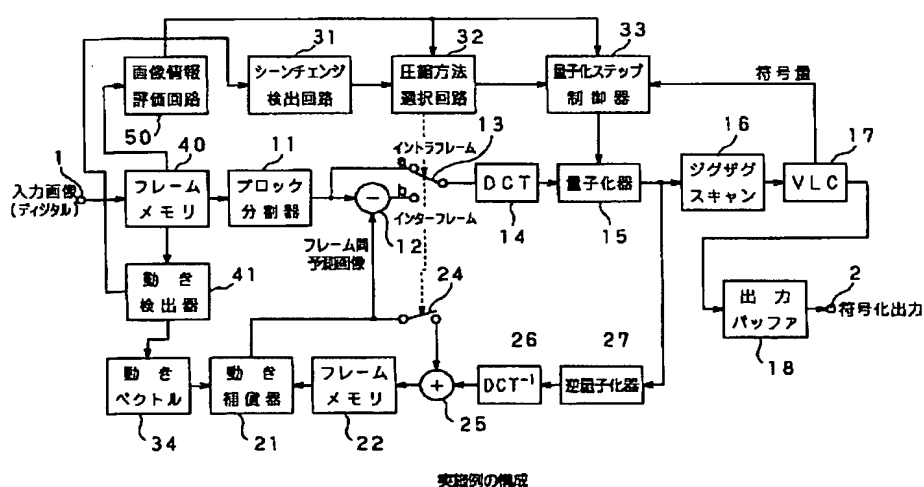
【図14】参照画像の構成について説明するための図である。

【図15】GOPの一例について説明するための図である。

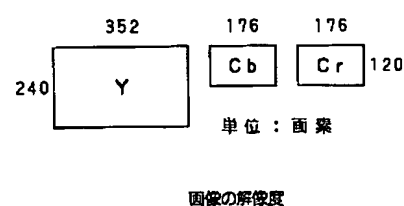
【符号の説明】

- 22 フレームメモリ
- 11 ブロック分割器
- 12 差分器
- 13, 24 スイッチ
- 14 DCT回路
- 15 量子化器
- 16 ジグザグスキャン回路
- 17 可変長符号化回路
- 18 出力バッファ
- 21 動き補償器
- 25 加算器
- 26 逆DCT回路
- 27 逆量子化器
- 31 シーンチェンジ検出回路
- 32 圧縮方法選択回路
- 33 量子化ステップ制御器
- 34 動きベクトル発生回路
- 41 動き検出器
- 50 画像情報評価回路

【図1】

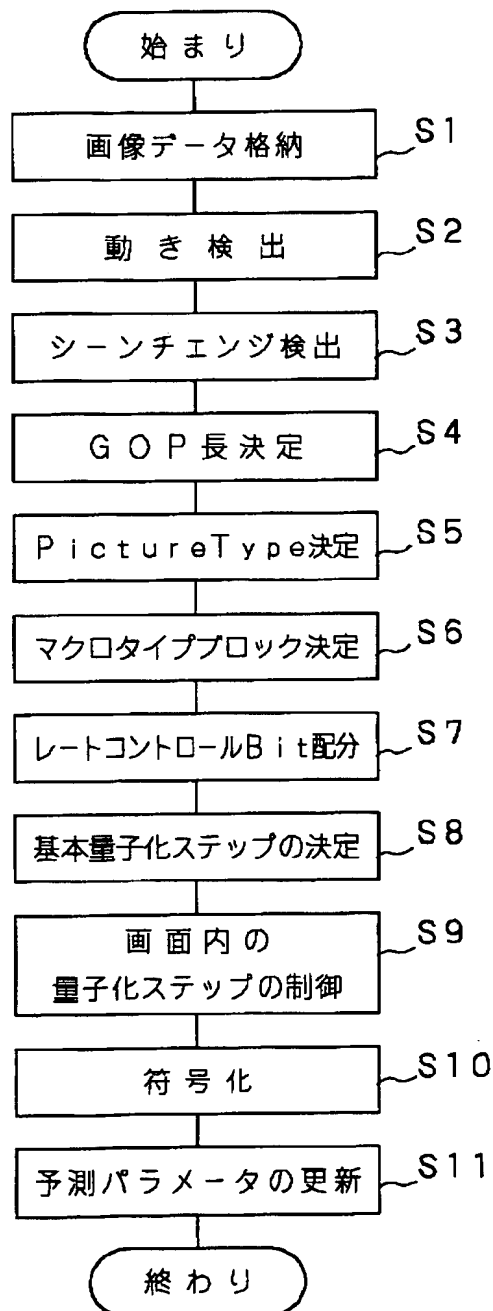


【図10】



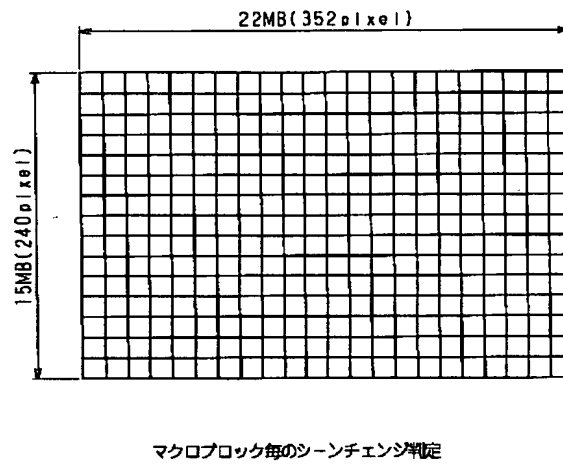
(15)

【図2】

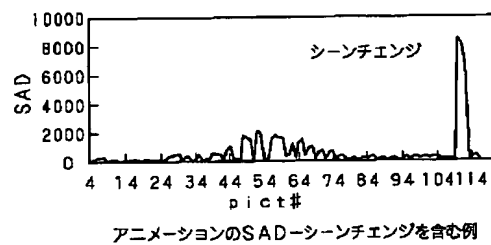


エンコードのフローチャート

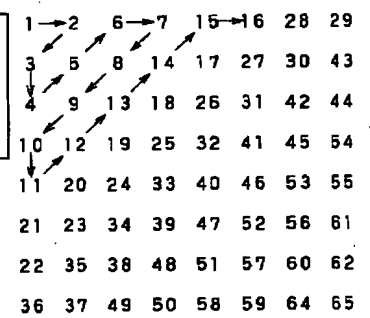
【図4】



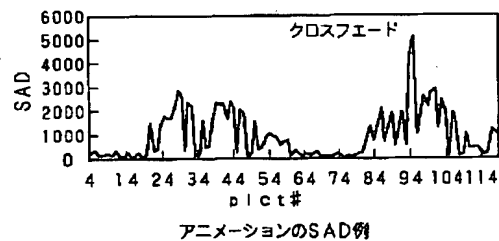
【図6】



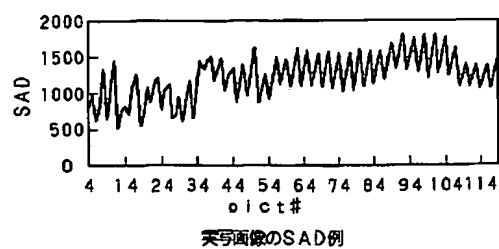
【図13】



【図7】

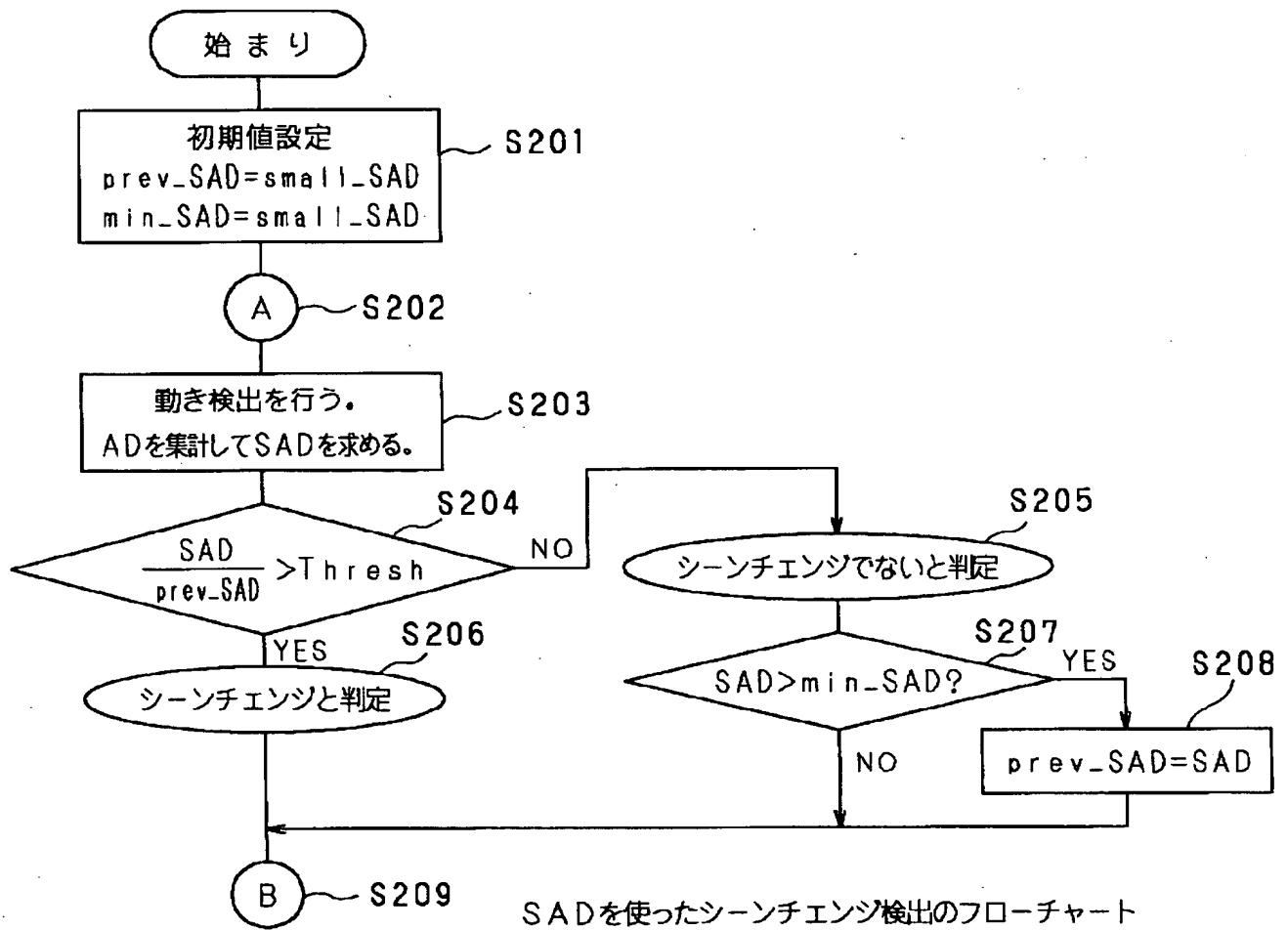


【図8】

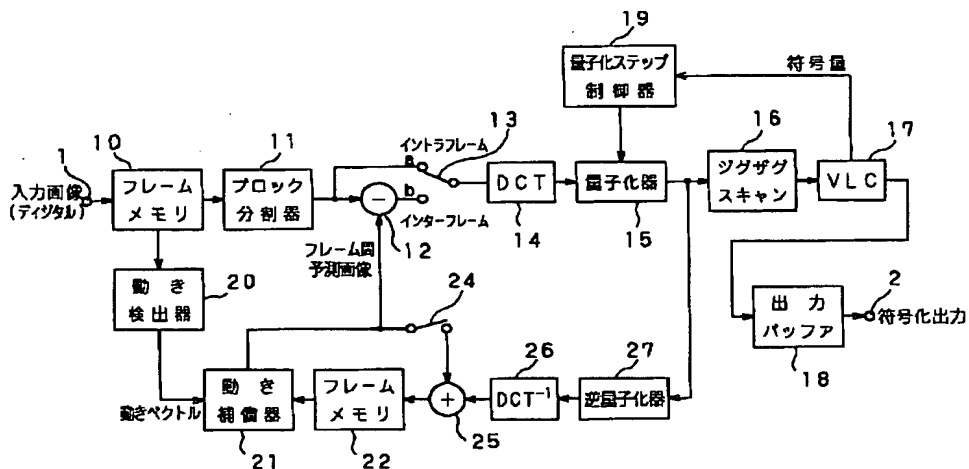


(16)

【図3】



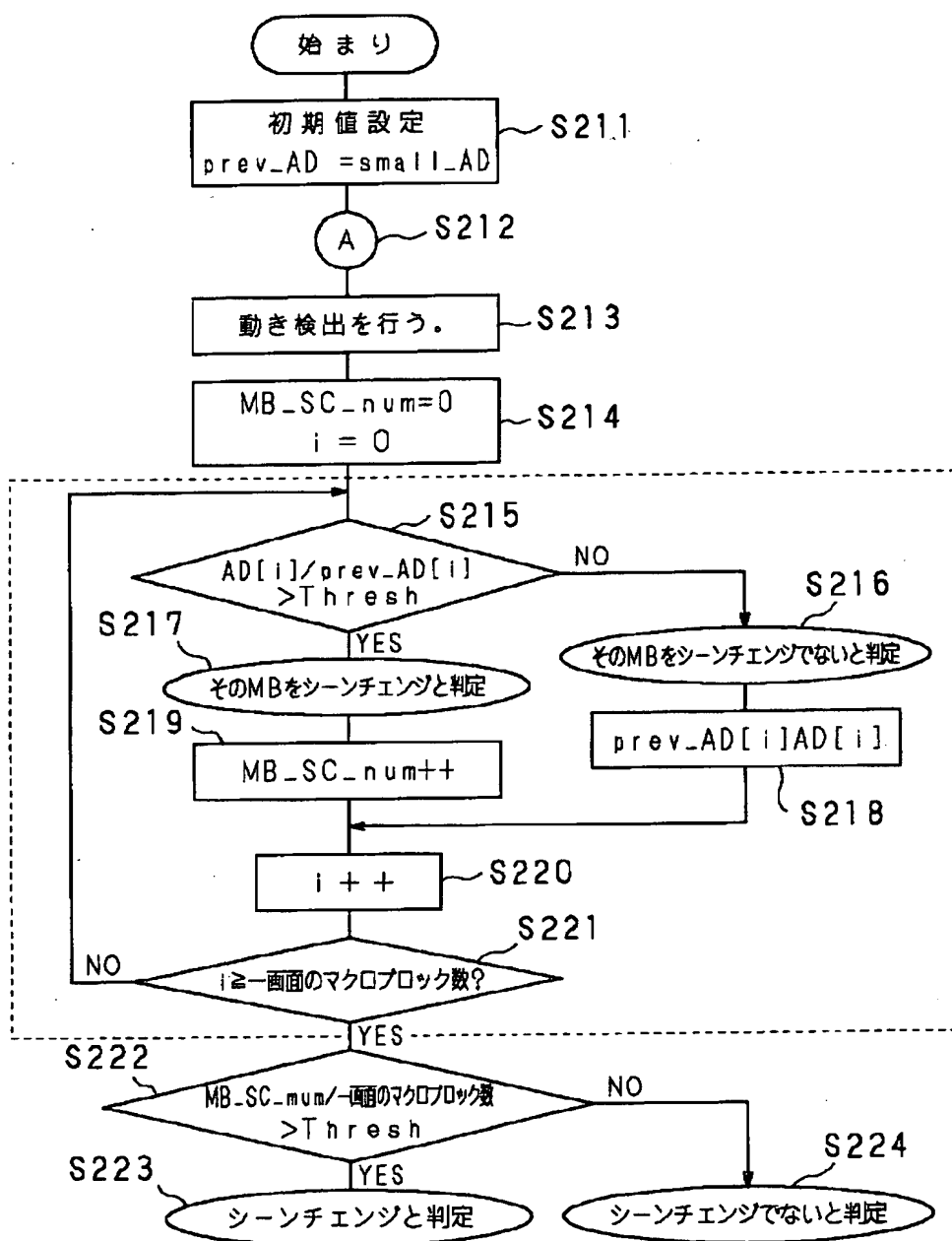
【図9】



従来例の構成

(17)

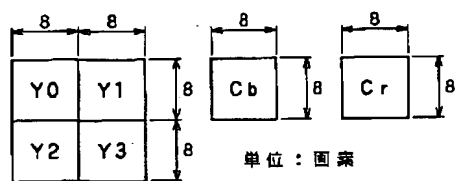
【図5】



ADを使ったMB毎比較判定シーンチェンジ検出のフローチャート

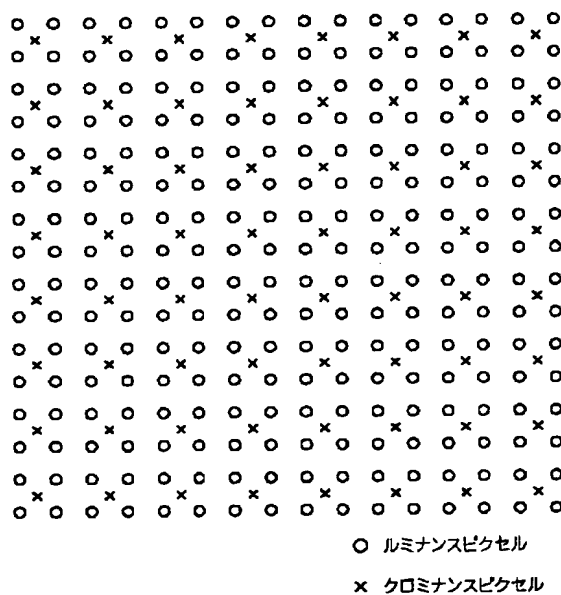
(18)

【図11】



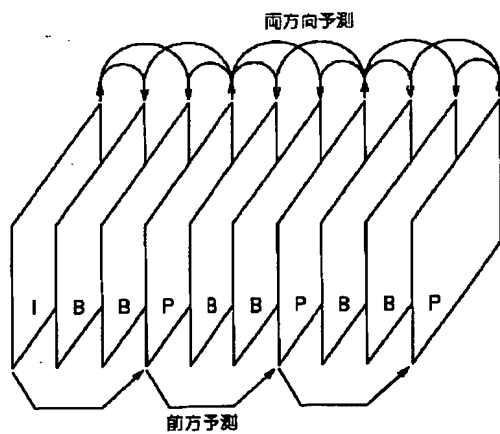
MB (マクロブロック) とブロック

【図12】



ルミナンスピクセルとクロミナンスピクセル

【図14】



動画画像の構成例

【図15】

